(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 1 154 348 A2

# (12) EUROPEAN PATENT APPLICATION

(43) Date of publication:
**14.11.2001 Bulletin 2001/46**

(51) Int Cl.7: **G06F 1/00**

(21) Application number: **01304182.7**

(22) Date of filing: **09.05.2001**

(84) Designated Contracting States:
**AT BE CH CY DE DK ES FI FR GB GR IE IT LI LU MC NL PT SE TR**
Designated Extension States:
**AL LT LV MK RO SI**

(30) Priority: **11.05.2000 JP 2000138642**

(71) Applicant: **Matsushita Electronics Corporation**
**Kadoma-shi, Osaka 571-8501 (JP)**

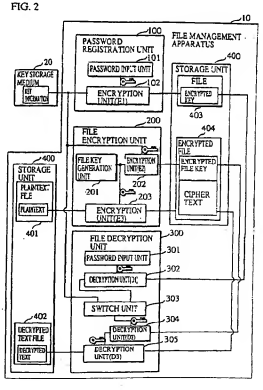
(72) Inventors:
- **Matsuzaki, Natsume**
  **Minou-shi, Osaka-fu 562-0023 (JP)**
- **Emura, Satoshi**
  **Toyonaka-shi, Osaka-fu 560-0034 (JP)**
- **Inagaki, Saturu**
  **Suita-shi, Osaka-fu 564-0063 (JP)**

(74) Representative: **Crawford, Andrew Birkby et al**
**A.A. Thornton & Co.**
**235 High Holborn**
**London WC1V 7LE (GB)**

## (54) File management apparatus

(57) A password registration unit encrypts key information using an input password, and stores the generated encrypted key as a file into a computer. A file encryption unit generates a file key arbitrarily, encrypts the file key using the key information, encrypts a plaintext using the file key to generate a ciphertext, and stores an encrypted file including the encrypted file key in its header part and the ciphertext in its data part. A file decryption unit decrypts the encrypted file key using the key information to obtain a file key, or receives an input of a password, decrypts the encrypted key using the password to obtain key information, and decrypts the encrypted file key using the key information to obtain a file key. The file decryption unit then decrypts the ciphertext using the obtained file key.

FIG. 2




EP 1 154 348 A2

## Description

**[0001]** This application is based on an application No. 2000-138642 filed in Japan, the content of which is hereby incorporated by reference.

## BACKGROUND OF THE INVENTION

### (1) Field of the Invention

**[0002]** The present invention relates to a file management apparatus that encrypts and stores information, to prevent third parties from knowing its contents.

### (2) Related Art

**[0003]** With the widespread use of computers, techniques for storing information after encrypting the information have been generally employed to prevent third parties from knowing the contents of the information.

**[0004]** Japanese Laid-Open Patent Application No. H9-204330 discloses a technique for encrypting a file in a computer using an encryption key and storing the encrypted file in a specific encrypted information storage area, to allow only specific users to have access to the encrypted information storage area with registered authentication passwords. Each specific user memorizes an authentication password. When the user inputs the authentication password, a decryption key is automatically selected so as to decrypt the encrypted file. Here, the authentication password may be composed of a character string or a number that is short enough for a person to memorize, and the encryption key and the decryption key have more bits than the authentication password.

**[0005]** According to the above technique, however, the difficulty still lies in that the user has to memorize the authentication password. In case the user forgets the authentication password, he or she cannot decrypt the encrypted file

## SUMMARY OF THE INVENTION

**[0006]** In view of the above problem, the object of the present invention is to provide a file management apparatus that is capable of managing encrypted information securely, and that ensures decryption of the encrypted information even when the user forgets a password.

**[0007]** The above object can be achieved by a file management apparatus that encrypts a plaintext to generate a ciphertext, stores the ciphertext, and decrypts the ciphertext, the file management apparatus including: a key storage medium storing key information beforehand; a registration unit for encrypting the key information using a password to generate an encrypted key; an encryption unit for encrypting a plaintext based on the key information to generate a ciphertext; a switch unit for switching between (a) generating key information by decrypting the encrypted key using the password and (b) reading the key information from the key storage medium; and a decryption unit for decrypting the ciphertext based on one of the generated key information and the read key information. The file management apparatus may further include a memory unit, wherein the registration unit receives an input of the password, encrypts the key information using the received password to generate the encrypted key, and writes the generated encrypted key to the memory unit, the encryption unit encrypts the plaintext using a file key to generate the ciphertext, encrypts the file key using the key information to generate an encrypted file key, and writes the ciphertext in association with the encrypted file key, to the memory unit, the switch unit (a) includes a first key obtaining unit for receiving an input of the password and decrypting the encrypted key using the received password to generate the key information, and a second key obtaining unit for reading the key information from the key storage medium, and (b) obtains the key information by one of the first key obtaining unit and the second key obtaining unit, and the decryption unit decrypts the encrypted file key using the obtained key information to generate a file key, and decrypts the ciphertext using the file key to generate a decrypted text.

**[0008]** According to this construction, operations are switched between (a) generating key information by decrypting the encrypted key using the password and (b) reading key information from the key storage medium, and the ciphertext is decrypted based on the generated key information or the read key information. Therefore, the ciphertext can be decrypted without a password.

**[0009]** The above object can also be achieved by a file management apparatus that encrypts a plaintext to generate a ciphertext, stores the ciphertext, and decrypts the ciphertext, the file management apparatus including: a key storage medium storing key information beforehand; a registration unit for encrypting a password using the key information to generate an encrypted password; an encryption unit for encrypting a plaintext using a file key to generate a ciphertext, encrypting the file key based on a password obtained by decrypting the encrypted password to generate a first encrypted file key, and encrypting the file key based on the key information to generate a second encrypted file key; a switch unit for switching between (a) decrypting the first encrypted file key based on the password and (b) decrypting the second encrypted file key based on the key information, to generate a file key; and a decryption unit for decrypting the ciphertext using the generated file key.

The file management apparatus may further include a memory unit, wherein the registration unit receives an input of the password, encrypts the received password using the key information to generate the encrypted password, and writes the generated encrypted password to the memory unit, the encryption unit decrypts the encrypted password using the key information to

generate the password, encrypts the plaintext using the file key to generate the ciphertext, encrypts the file key using the password to generate the first encrypted file key, encrypts the file key using the key information to generate the second encrypted file key, and writes the ciphertext in association with the first encrypted file key and the second encrypted file key, to the memory unit, the switch unit (a) includes a first key obtaining unit for receiving an input of the password and decrypting the first encrypted file key using the received password, and a second key obtaining unit for decrypting the second encrypted file key using the key information, and (b) obtains the file key by one of the first key obtaining unit and the second key obtaining unit, and the decryption unit decrypts the ciphertext using the obtained file key to generate a decrypted text.

**[0010]** According to this construction, operations are switched between (a) decrypting the encrypted file key based on the password and (b) decrypting an encrypted file key based on the key information, to generate a file key, and the ciphertext is decrypted based on the file key. Therefore, the ciphertext can be decrypted without a password.

## BRIEF DESCRIPTION OF THE DRAWINGS

**[0011]** These and other objects, advantages and features of the invention will become apparent from the following description thereof taken in conjunction with the accompanying drawings that illustrate a specific embodiment of the invention. In the drawings:

Fig. 1 shows an appearance of a file management apparatus relating to a first embodiment of the present invention;
Fig. 2 is a block diagram showing a construction of the file management apparatus;
Fig. 3 is a flowchart showing an operation of a password registration unit in the first embodiment;
Fig. 4 is a flowchart showing an operation of a file encryption unit in the first embodiment;
Fig. 5 is a flowchart showing an operation of a file decryption unit in the first embodiment;
Fig. 6 shows an example of a user ID table;
Fig. 7 is a flowchart showing an operation of the file management apparatus when a password is changed;
Fig. 8 is a flowchart showing an operation of the file management apparatus when key information is changed;
Fig. 9 shows an example of data structure of an encrypted file in the first embodiment;
Fig. 10 is a block diagram showing a construction of a file apparatus relating to a second embodiment of the present invention;
Fig. 11 is a flowchart showing an operation of a password registration unit in the second embodiment;
Fig. 12 is a flowchart showing an operation of a file encryption unit in the second embodiment;
Fig. 13 is a flowchart showing an operation of a file decryption unit in the second embodiment;
Fig. 14 is a flowchart showing an operation of the file management apparatus when a password is changed;
Fig. 15 is a flowchart showing an operation of the file management apparatus when key information is changed;
Fig. 16 is a flowchart showing an operation when a key storage medium is lost in the second embodiment. To be i continued to Fig. 17;
Fig. 17 is a flowchart showing the operation when the key storage medium is lost in the second embodiment. To be continued to Fig. 18; and
Fig. 18 is a flowchart showing the operation when the key storage medium is lost in the second embodiment. Continued from Fig. 17.

## DESCRIPTION OF THE PREFERRED EMBODIMENTS

**[0012]** The following is an explanation of preferred embodiments of the present invention, with reference to the drawings.

### 1. First Embodiment

**[0013]** The following is an explanation of a file management apparatus 10 relating to a first embodiment of the present invention.

**[0014]** Fig. 1 shows an appearance of the file management apparatus 10. As shown in the figure, the file management apparatus 10 is a computer system that is roughly composed of a microprocessor, a ROM, a RAM, a hard disc unit, a display unit, and a keyboard. The RAM or the hard disk unit stores a computer program. The functions of the file management apparatus 10 are realized by the microprocessor operating according to the computer program. A key storage medium 20 which stores key information beforehand is equipped in the file management apparatus 10.

### 1.1 Constructions of the File Management Apparatus 10 and the Key Storage Medium 20

**[0015]** The following is an explanation of the constructions of the file management apparatus 10 and the key storage medium 20.

**[0016]** As shown in Fig. 2, the file management apparatus 10 includes a password registration unit 100, a file encryption unit 200, a file decryption unit 300, and a storage unit 400, and the key storage medium 20 is connected to the file management apparatus 10.

**[0017]** The password registration unit 100 includes a password input unit 101 and an encryption unit 102. The file encryption unit 200 includes a file key generation unit

201, an encryption unit 202, and an encryption unit 203. The file decryption unit 300 includes a password input unit 301, a decryption unit 302, a switch unit 303, a decryption unit 304, and a decryption unit 305.

(1) Key Storage Medium 20

**[0018]** The key storage medium 20 is a portable storage medium having a storage area made up of a non-volatile semiconductor memory. The storage area stores 56-bit key information beforehand.

**[0019]** The key information is unique to a user, and the user usually possesses the key storage medium 20. To operate the file management apparatus 10, the user inserts the key storage medium 20 in a special drive equipped with the file management apparatus 10, to connect the key storage medium 20 to the file management apparatus 10.

(2) Storage Unit 400

**[0020]** The storage unit 400 is constructed of a hard disc unit, and is internally equipped with a storage area for storing information as files. Each file is identified by a file name.

**[0021]** The storage unit 400 stores a plaintext file 401 beforehand, the plaintext 401 storing a plaintext.

(3) Password Input Unit 101

**[0022]** The password input unit 101 receives an input of a password from the user. Here, the password is a string of eight characters composed of numerals and alphabets. The password input unit 101 outputs the received password to the encryption unit 102.

(4) Encryption Unit 102

**[0023]** The encryption unit 102 receives the password from the password input unit 101. On receipt of the password, the encryption unit 102 reads the key information from the storage area of the key storage medium 20, adds a plurality of zero bits to the end of the password to make it 56 bits long, and adds a plurality of zero bits to the end of the key information to make the key information 64 bits long. Following this, the encryption unit 102 subjects the key information to the encryption algorithm E1 using the password as a key to generate an encrypted key. Here, the encryption algorithm E1 complies with Data Encryption Standard (DES). Note that DES is well-known, and so it is not explained here.

**[0024]** In a block diagram in Fig. 2, a key mark near a line connecting the password input unit 101 and the encryption unit 102 indicates that the encryption unit 102 uses the password outputted from the password input unit 101 as a key. The same applies to other encryption and decryption units in Fig. 2, and to encryption and decryption units in Fig. 10.

**[0025]** The encryption unit 102 then writes the generated encrypted key as a file to the storage unit 400.

(5) File Key Generation Unit 201

**[0026]** The file key generation unit 201 is internally equipped with a random number generation unit and a timer, and so generates a 56-bit random number, acquires the current time expressed by year, month, day, hour, minute, second, and millisecond, takes an exclusive-OR of the generated random number and the acquired current time so as to generate a file key that is 56 bits long, and outputs the generated file key to the encryption unit 202 and the encryption unit 203.

(6) Encryption Unit 203

**[0027]** The encryption unit 203 receives the user designation of a file name of the plaintext file 401 stored in the storage unit 400, and reads the plaintext file 401 identified by the file name from the storage unit 400. Also, the encryption unit 203 receives the file key from the file key generation unit 201.

**[0028]** The encryption unit 203 then subjects a plaintext included in the plaintext file 401 to the encryption algorithm E3 using the received file key as a key, to generate a ciphertext. The encryption unit 203 then writes an encrypted file 404 to the storage unit 400. The encrypted file 404 is composed of a header part, and a data part that includes the generated ciphertext. It should be noted here that the encryption algorithm E3 complies with DES.

**[0029]** Here, when the plaintext is at least 64 bits long, the encryption unit 203 divides the plaintext into a plurality of plaintext blocks, each plaintext block being 64 bits long. The encryption unit 203 then subjects each plaintext block to the encryption algorithm E3 to generate a ciphertext block, and concatenates each generated ciphertext block to form a ciphertext.

(7) Encryption Unit 202

**[0030]** The encryption unit 202 reads the key information from the key storage medium 20, receives the file key from the file key generation unit 201, and adds a plurality of zero bits to the end of the file key so as to make the file key 64 bits long.

**[0031]** The encryption unit 202 then subjects the file key to the encryption algorithm E2 using the read key information as a key to generate an encrypted file key, and writes the generated encrypted file key into the header part of the encrypted file 404 in the storage unit 400. It should be noted here that the encryption algorithm E2 complies with DES.

(8) Switch Unit 303

**[0032]** The switch unit 303 receives an input of either

a first type or a second type from the user. The first type indicates to decrypt a ciphertext using a password, and the second type indicates to decrypt a ciphertext using key information.

**[0033]** When the input of the first type is received, the switch unit 303 receives the key information from the decryption unit 302, and outputs the received key information to the decryption unit 304. When the input of the second type is received, the switch unit 303 reads the key information from the key storage medium 20, and outputs the read key information to the decryption unit 304.

(9) Password Input Unit 301

**[0034]** The password input unit 301, as the password input unit 101, receives the input of the password from the user and outputs the received password to the decryption unit 302.

(10) Decryption Unit 302

**[0035]** The decryption unit 302 receives the password from the password input unit 301, reads the encrypted key from the storage unit 400, adds a plurality of zero bits to the end of the password so as to make the password 56 bits long, and subjects the read encrypted key to the decryption algorithm D1 using the password as a key to generate key information. It should be noted here that the decryption algorithm D1 complies with DES, and is to perform the inverse conversion to the encryption algorithm E1.

**[0036]** Following this, the decryption unit 302 deletes the bit string of the generated key information except the first 56 bits, and outputs the 56-bit key information to the switch unit 303.

(11) Decryption Unit 304

**[0037]** The decryption unit 304 receives the key information from the switch unit 303, reads the encrypted file key included in the header part of the encrypted file 404 in the storage unit 400, and subjects the read encrypted file key to the decryption algorithm D2 using the received key information as a key to generate a file key. It should be noted here that the decryption algorithm D2 complies with DES, and is to perform the inverse conversion to the encryption algorithm E2.

**[0038]** The decryption unit 304 then deletes the bit string of the generated file key except the first 56 bits, and outputs the 56-bit file key to the decryption unit 305.

(12) Decryption Unit 305

**[0039]** The decryption unit 305 receives the file key from the decryption unit 304, reads the ciphertext included in the data part of the encrypted file 404 in the storage unit 400, and subjects the read ciphertext to the decryption algorithm D3 using the received file key as a key to generate a decrypted text. It should be noted here that the decryption algorithm D3 complies with DES, and is to perform the inverse conversion to the encryption algorithm E3.

**[0040]** Here, when the ciphertext is at least 64 bits long, the decryption unit 305 divides the ciphertext into a plurality of ciphertext blocks, each ciphertext block being 64 bits long. The decryption unit 305 then subjects each ciphertext block to the decryption algorithm D3 to generate a decrypted text block, and concatenates each generated decrypted text block to form a decrypted text.

**[0041]** Following this, the decryption unit 305 writes a decrypted text file 402 including the generated decrypted text to the storage unit 400.

1.2 Operation of the File Management Apparatus 10

**[0042]** The following is an explanation of the operation of the file management apparatus 10.

(1) Operation of the Password Registration Unit 100

**[0043]** The following is an explanation of the operation of the password registration unit 100, with reference to a flowchart shown in Fig. 3.

**[0044]** The password input unit 101 receives an input of a password from the user, and outputs the received password to the encryption unit 102 (step S101).

**[0045]** The encryption unit 102 then reads key information from the storage area of the key storage medium 20 (step S102), subjects the read key information to the encryption algorithm E1 using the password as a key to generate an encrypted key (step S103), and writes the generated encrypted key as a file to the storage unit 400 (step S104).

(2) Operation of the File Encryption Unit 200

**[0046]** The following is an explanation of the operation of the file encryption unit 200, with reference to a flowchart shown in Fig. 4.

**[0047]** The file key generation unit 201 generates a file key (step S121). Following this, the encryption unit 203 reads the plaintext file 401 from the storage unit 400, subjects a plaintext stored in the plaintext file 401 to the encryption algorithm E3 using the generated file key as a key to generate a ciphertext (step S122), and writes the encrypted file 404 including the generated ciphertext in the data part thereof, to the storage unit 400 (step S123).

**[0048]** Following this, the encryption unit 202 reads key information from the key storage medium 20, receives the file key from the file key generation unit 201, subjects the received file key to the encryption algorithm E2 using the read key information as a key to generate an encrypted file key (step S124), and writes the generated encrypted file key into the header part of the en-

crypted file 404 in the storage unit 400 (step S125).

(3) Operation of the File Decryption Unit 300

**[0049]** The following is an explanation of the operation of the file decryption unit 300, with reference to a flowchart shown in Fig. 5.

**[0050]** The switch unit 303 receives an input of either the first type or the second type from the user (step S141).

**[0051]** When the switch unit 303 receives the input of the first type (step S142), the password input unit 301 receives an input of a password from the user and outputs the received password to the decryption unit 302 (step S144). The decryption unit 302 reads an encrypted key from the storage unit 400, subjects the read encrypted key to the decryption algorithm D1 using the password as a key to generate key information, and outputs the generated key information to the decryption unit 304 via the switch unit 303 (step S145).

**[0052]** When the switch unit 303 receives the input of the second type (step S142), the switch unit 303 reads key information from the key storage medium 20, and outputs the read key information to the decryption unit 304 (step S143).

**[0053]** Following this, the decryption unit 304 receives the key information from the switch unit 303, reads an encrypted file key included in the header part of the encrypted file 404 in the storage unit 400, and subjects the read encrypted file key to the decryption algorithm D2 using the received key information as a key to generate a file key (step S146). The decryption unit 305 reads a ciphertext included in the data part of the encrypted file 404 in the storage unit 400, subjects the read ciphertext to the decryption algorithm D3 using the file key as a key to generate a decrypted text (step S147), and writes the decrypted text file 402 including the generated decrypted text, to the storage unit 400 (step S148).

1.3 Conclusions

**[0054]** As described above, the file management apparatus 10 has the three functions : password registration; plaintext encryption; and ciphertext decryption.

**[0055]** For registering a password, the user loads the key storage medium 20 on the file management apparatus 10, and inputs a password to be registered. The password registration unit 100 encrypts key information using the input password, and stores the generated encrypted key as a file in the computer.

**[0056]** For encrypting a plaintext, the user loads the key storage medium 20 on the file management apparatus 10, and designates a file to be encrypted. Here, a password does not need to be inputted for encrypting each plaintext, which makes the encryption processing easier for the user. The file encryption unit 200 generates a file key arbitrarily, encrypts the generated file key using the key information to generate an encrypted file key, encrypts information stored in the file using the generated file key to generate a ciphertext, and writes an encrypted file to the storage unit 400, the encrypted file including the encrypted file key in the header part thereof and the ciphertext in the data part thereof.

**[0057]** For decrypting a ciphertext, there are two methods, one using key information and the other using a password. When using key information, the file decryption unit 300 decrypts an encrypted file key obtained from the header part of the encrypted file using the key information, to obtain a file key. The file decryption unit 300 then decrypts a ciphertext using the obtained file key as a key. When using a password, the file decryption unit 300 receives an input of a password from the user, decrypts an encrypted key using the received password to obtain key information, decrypts an encrypted file key using the key information to obtain a file key, and finally decrypts a ciphertext using the file key as a key to obtain the plaintext.

**[0058]** According to the above construction of the file management apparatus 10, encrypted information is usually decrypted using key information, and when the user fails to bring a key storage medium storing key information, encrypted information can be decrypted using a password as described above.

1.4 Modifications

**[0059]** Although the present invention has been described based on the first embodiment, the invention should not be limited to such. For instance, the file management apparatus 10 may be constructed according to the following modifications.

**[0060]** (1) The password registration unit 100 may further receive an input of a user identifier (user ID) that identifies the user, and write the encrypted key, in association with the user identifier, into a user ID table in the storage unit 400. Fig. 6 shows an example of the user ID table. The user ID table has an area for storing a plurality of pairs each composed of an user ID and an encrypted key. In this case, the file decryption unit 300 receives an input of a user ID, and then decrypts an encrypted key that is associated with the input user ID in the user ID table.

**[0061]** With this construction, a plurality of users can use the file management apparatus 10.

**[0062]** (2) The following is an explanation of the operation of the file management apparatus 10 when a password is changed, with reference to a flowchart shown in Fig. 7.

**[0063]** The file management apparatus 10 further includes a deletion unit for deleting the encrypted key stored in the storage unit 400 (step S161).

**[0064]** The password input unit 101 in the password registration unit 100 receives an input of a new password from the user, and outputs the received new password to the encryption unit 102 (step S162). The encryption unit 102 then reads key information from the storage

area of the key storage medium 20 (step S163), subjects the read key information to the encryption algorithm E1 using the new password as a key, to obtain a new encrypted key (step S164), and writes the generated new encrypted key as a file to the storage unit 400 (step S165).

**[0065]** In the above described way, a new encrypted key is generated when the password is changed.

**[0066]** (3) For preventing encrypted information from being decrypted using a password, the only thing to do is to delete the encrypted key that has been encrypted using the password.

**[0067]** (4) The following is an explanation of the operation of the file management apparatus 10 when key information is updated, with reference to a flowchart shown in Fig. 8.

**[0068]** The key storage medium 20 stores new key information beforehand, instead of the key information employed previously (referred to as old key information).

**[0069]** The password input unit 101 receives an input of a password that is the same as the password received previously (step S181). The encryption unit 102 subjects the encrypted key (hereafter referred to as the old encrypted key) to the decryption algorithm D1 using the received password as a key to generate key information that is the same as the old key information (step S182), reads the new key information from the key storage medium 20, subjects the read new key information to the encryption algorithm E1 using the password as a key to generate a new encrypted key (step S183), and updates the old encrypted key stored in the storage unit 400 to the generated new encrypted key (step S184).

**[0070]** The file encryption unit 200 then reads the encrypted file key generated previously (hereafter referred to as the old encrypted file key) from the storage unit 400, and subjects the old encrypted file key to the decryption algorithm D2 using the old key information as a key, to generate a file key (step S185), reads the new key information from the key storage medium 20, subjects the file key to the encryption algorithm E2 using the new key information as a key to generate a new encrypted file key (step S186), and updates the old encrypted file key in the encrypted file to the new encrypted file key (step S187).

**[0071]** In this way, for updating key information, the key information before being updated is first obtained using the old encrypted key and the password. An encrypted file key included in the header is then decrypted using the old key information to obtain a file key. Following this, the file key is encrypted using the new key information, and the encrypted file key is updated. Here, the encrypted key is updated, too.

**[0072]** Note in the present embodiment, when key information is lost, the key information cannot be made temporarily invalid.

**[0073]** (5) When encrypting a plaintext, the file encryption unit 200 may add encryption information to the header part of the encrypted file, the encryption information indicating that the plaintext has been encrypted. In this case, when key information is updated, the file encryption unit 200 may retrieve the encrypted file key in the encrypted file 404 to whose header the encryption information has been added, and generate a file key from the retrieved encrypted file key.

**[0074]** Also, the password registration unit 100 may receive an input of a user ID that identifies the user, and the file encryption unit 200 may additionally write the user ID to the encrypted file that includes the ciphertext and the encrypted file key. In this case, when key information is updated, the file encryption unit 200 may retrieve the encrypted file key in the encrypted file to which the user ID has been added, and generate a file key from the retrieved encrypted file key.

**[0075]** Also, the file encryption unit 200 may write the user ID and a file identifier that identifies the encrypted file including the ciphertext and the encrypted file key, in association with each other, as a unified file, to the storage unit 400. In this case, the file encryption unit 200 may extract the file identifier that is associated with the user ID from the unified file, identify the encrypted file key included in the file identified by the extracted file identifier, and generate a file key from the identified encrypted file key.

**[0076]** Alternatively, the file encryption unit 200 may write (a) encryption information indicating that the plaintext has been encrypted and (b) a file identifier that identifies the encrypted file including the ciphertext and the encrypted file key, in association with each other, as a unified file, to the storage unit 400. In this case, the file encryption unit 200 may extract the file identifier that is associated with the encryption information from the unified file, identify the encrypted file key included in the file identified by the extracted file identifier, and generate a file key from the identified encrypted file key.

**[0077]** (6) In the above embodiment, the encrypted key is stored in one computer system, and so decryption of a ciphertext using a password is made only possible within the computer system. To enable the decryption of the ciphertext using the password in another computer system, the encrypted key may be stored in a portable storage medium, and may be inputted into the other computer system.

**[0078]** Here, the password registration unit 100 in the computer system writes the encrypted key to a portable storage medium such as a SD memory card. Also, the user writes the encrypted file to another portable storage medium. The user then loads the portable storage medium to which the encrypted key has been written, and the portable storage medium to which the encrypted file has been written, on the other computer system, so that a file decryption unit in the other computer system reads the encrypted key from the portable storage medium, decrypts the read encrypted key, and also, reads the encrypted file from the portable storage medium, and decrypts the read encrypted file.

**[0079]** It should be noted here that the encrypted key and the encrypted file may be written to one portable storage medium as separate files.

**[0080]** (7) The password registration unit 100 may read key information from the key storage medium 20, subject the read key information to a hash algorithm to generate first authentication information, and write the generated first authentication information in association with the encrypted key, to the storage unit 400. In this case, the file decryption unit 300 may read the encrypted key and the first authentication information from the storage unit 400, decrypt the encrypted key to generate key information, and subject the generated key information to the hash algorithm that was used in the above encryption, to generate second authentication information. Following this, the file decryption unit 300 may compare the first authentication information and the second authentication information to see if they match. If they do not match, the encrypted key is judged to have been altered, or if they match, the encrypted key is judged not to have been altered.

**[0081]** The file encryption unit 200 may also generate first authentication information from a file key in the same way as described above, and writes the generated first authentication information in association with the encrypted file key, to the storage unit 400. The file decryption unit 300 may read the first authentication information and the file key, generate second authentication information from the read file key in the same way as described above, and compare the read first authentication information with the generated second authentication information, to detect an alteration of the file key if any. Also, an alteration of a plaintext can be detected in the same manner as described above.

**[0082]** (8) The password registration unit 100 may write the key information and the encrypted key, in association with each other, as one file to the storage unit 400.

**[0083]** As one example shown in Fig. 9, the file encryption unit 200 writes the encrypted key and the encrypted file key to the header part of the encrypted file 404a, and the ciphertext to the data part of the encrypted file 404a in the storage unit 400b. In this case, the file decryption unit 300 reads the encrypted key from the header part of the encrypted file 404a, instead of reading the encrypted key from the file 403 in the storage unit 400.

**[0084]** By storing the encrypted key to a header part of each encrypted file, a ciphertext stored therein can be decrypted only using a password if the encrypted file is transferred to another computer. It should be noted here, however, when the password is changed, the encrypted key in the header part of each concerned encrypted file needs to be updated. Also, storing the encrypted key and the key information required for encrypting a plaintext into one storage medium serves as convenient.

**[0085]** (9) The file encryption unit 200 may further receive an input of a user indication, the user indication showing whether an encrypted key and a ciphertext are to be stored in association with each other into one encrypted file. When the indication shows that the encrypted key and the ciphertext are to be stored in association with each other into one encrypted file, the file encryption unit 200 writes the encrypted key to the header part of the encrypted file, and the ciphertext to the data part of the encrypted file.

**[0086]** It should be noted here that an encrypted file that does not store an encrypted key cannot be decrypted only with a password unless the encrypted key is stored separately.

**[0087]** (10) The password registration unit 100 may write the generated encrypted key to the key storage medium 20 instead of to the storage unit 400.

## 2. Second Embodiment

**[0088]** The following is an explanation of a file management apparatus 10b relating to a second embodiment of the present invention.

**[0089]** The file management apparatus 10b is a computer system on which the key storage medium 20 is loaded, as the file management apparatus 10.

### 2.1 Constructions of the File Management Apparatus 10b and the Key Storage Medium 20

**[0090]** The following is an explanation of the constructions of the file management apparatus 10b and the key storage medium 20.

**[0091]** The file management apparatus 10b includes a password registration unit 100b, a file encryption unit 200b, a file decryption unit 300b, and a storage unit 400b, and the key storage medium 20 is connected to the file management apparatus 10b as shown in Fig. 10.

**[0092]** The password registration unit 100b includes a password input unit 101b and an encryption unit 102b. The file encryption unit 200b includes a file key generation unit 201b, an encryption unit 202b, an encryption unit 203b, an encryption unit 204b, and a decryption unit 205b. The file decryption unit 300b includes a password input unit 301b, a decryption unit 302b, a switch unit 303b, a decryption unit 304b, and a decryption unit 305b. The following explanation focuses on the differences from the construction of the file management apparatus 10.

#### (1) Storage Unit 400b

**[0093]** The storage unit 400b, as the storage unit 400, stores a plaintext file 401b beforehand, the plaintext file 401b storing a plaintext.

#### (2) Password Input Unit 101b

**[0094]** The password input unit 101b, as the pass-

word input unit 101, receives an input of a password, and outputs the received password to the encryption unit 102b.

(3) Encryption Unit 102b

**[0095]** The encryption unit 102b, as the encryption unit 102, reads key information from the key storage medium 20, subjects the password received from the password input unit 101b to the encryption algorithm E1 using the read key information to generate an encrypted password, and writes the generated encrypted password as a file, to the storage unit 400b.

(4) File Key Generation Unit 201b

**[0096]** The file key generation unit 201b, as the file key generation unit 201, generates a file key, and outputs the generated file key to the encryption unit 202b, the encryption unit 203b, and the encryption unit 204b.

(5) Decryption Unit 205b

**[0097]** The decryption unit 205b reads the encrypted password stored in the storage unit 400b, and reads the key information from the key storage medium 20. The decryption unit 205b then subjects the read encrypted password to the decryption algorithm D1 using the read key information to generate a password, and outputs the generated password to the encryption unit 202b.

(6) Encryption Unit 203b

**[0098]** The encryption unit 203b, as the encryption unit 203, reads the plaintext file 401b from the storage unit 400b, and receives the file key from the file key generation unit 201b.

**[0099]** The encryption unit 203b then subjects a plaintext included in the plaintext file 401b to the encryption algorithm E3 using the received file key as a key to generate a ciphertext, and writes an encrypted file 404b including the generated ciphertext in the data part thereof, to the storage unit 400.

(7) Encryption Unit 202b

**[0100]** The encryption unit 202b receives the password from the decryption unit 205b and the file key from the file key generation unit 201b. The encryption unit 202b then subjects the received file key to the encryption algorithm E2 using the received password as a key to generate a first encrypted file key, and writes the generated first encrypted file key to the header part of the encrypted file 404b in the storage unit 400b.

(8) Encryption Unit 204b

**[0101]** The encryption unit 204b reads the key information from the key storage medium 20, receives the file key from the file key generation unit 201b. The encryption unit 204b then subjects the file key to the encryption algorithm E4 using the read key information as a key to generate a second encrypted file key, and writes the generated second encrypted file key to the header part of the encrypted file 404b in the storage unit 400b. It should be noted here that the encryption algorithm E4 complies with DES.

(9) Switch Unit 303b

**[0102]** The switch unit 303b receives an input of either a first type or a second type from the user. The first type indicates to decrypt a ciphertext using a password, and the second type indicates to decrypt a ciphertext using key information.

**[0103]** When the input of the first type is received, the switch unit 303b receives the file key from the decryption unit 302b, and outputs the received file key to the decryption unit 305b. When the input of the second type is received, the switch unit 303b receives the file key from the decryption unit 304b, and outputs the received file key to the decryption unit 305b.

(10) Password Input Unit 301b

**[0104]** The password input unit 301b, as the password input unit 101, receives an input of a password from the user, and outputs the received password to the decryption unit 302b.

(11) Decryption Unit 302b

**[0105]** The decryption unit 302b receives the password from the password input unit 301b, reads the first encrypted file key included in the header part of the encrypted file 404b in the storage unit 400b. The decryption unit 302b then subjects the read first encrypted file key to the decryption algorithm D2 using the read password as a key to generate a file key, and outputs the generated file key to the switch unit 303b.

(12) Decryption Unit 304b

**[0106]** The decryption unit 304b reads the key information from the key storage medium 20, reads the second encrypted file key included in the header part of the encrypted file 404 in the storage unit 400b, and subjects the read second encrypted file key to the decryption algorithm D4 using the read key information as a key to generate a fie key. Here, the decryption algorithm D4 complies with DES, and is to perform the inverse conversion to the encryption algorithm E4. The decryption unit 304b outputs the generated file key to the switch unit 303b.

(13) Decryption Unit 305b

**[0107]** The decryption unit 305b receives the file key from the decryption unit 304b, reads a ciphertext included in the data part of the encrypted file 404b in the storage unit 400, and subjects the read ciphertext to the decryption algorithm D3 using the received file key as a key to generate a decrypted text. The decryption unit 305b writes a decrypted text file 402b including the generated decrypted text to the storage unit 400.

2.2 Operation of the File Management Apparatus 100b

**[0108]** The following is an explanation of the operation of the file management apparatus 10b.

(1) Operation of the Password Registration Unit 100b

**[0109]** The following is an explanation of the operation of the password registration unit 100b, with reference to a flowchart shown in Fig. 11.

**[0110]** The password input unit 101b receives an input of a password from the user, and outputs the received password to the encryption unit 102b (step S201).

**[0111]** The encryption unit 102b then reads key information from the storage area of the key storage medium 20 (step S202), subjects the password to the encryption algorithm E1 using the key information as a key to generate an encrypted password (step S203), and writes the generated encrypted password as a file, to the storage unit 400b (step S204).

(2) Operation of the File Encryption Unit 200b

**[0112]** The following is an explanation of the operation of the file encryption unit 200b, with reference to a flowchart shown in Fig. 12.

**[0113]** The decryption unit 205b reads an encrypted password stored in the storage unit 400b, reads key information from the key storage medium 20, subjects the read encrypted pas sword to the decryption algorithm D1 using the read key information to generate a password, and writes the generated password to the encryption unit 202b (step S221).

**[0114]** Following this, the file key generation unit 201b generates a file key (step S222).

**[0115]** The encryption unit 203b then reads the plaintext file 401b from the storage unit 400b, subjects a plaintext included in the plaintext file 401b to the encryption algorithm E3 using the file key as a key to generate a ciphertext (step S223), and writes the encrypted file 404b including the generated ciphertext in the data part thereof, to the storage unit 400b (step S224).

**[0116]** Following this, the encryption unit 202b receives the password and the file key, and subjects the file key to the encryption algorithm E2 using the password as a key to generate a first encrypted file key (step S225), and writes the generated first encrypted file key to the header part of the encrypted file 404b in the storage unit 400b (step S226).

**[0117]** Following this, the encryption unit 204b receives the file key and the key information, subjects the file key to the encryption algorithm E4 using the key information as a key to generate a second encrypted file key (step S227), and writes the generated second encrypted file key to the header part of the encrypted file 404b in the storage unit 400b (step S228).

(3) Operation of the File Decryption Unit 300b

**[0118]** The following is an explanation of the operation of the file decryption unit 300b, with reference to a flowchart shown in Fig. 13.

**[0119]** The switch unit 303b receives an input of either the first type or the second type from the user (step S241).

**[0120]** When the switch unit 303b receives the input of the first type (step S242), the password input unit 301b receives an input of a password from the user and outputs the received password to the decryption unit 302b (step S245). The decryption unit 302b reads a first encrypted file key from the storage unit 400b, subjects the read first encrypted file key to the decryption algorithm D2 using the password as a key to generate a file key, and outputs the generated file key to the decryption unit 305b via the switch unit 303b (step S246).

**[0121]** When the switch unit 303b receives the input of the second type (step S242), the decryption unit 304b reads key information from the key storage medium 20 (step S243), reads a second encrypted file key from the storage unit 400b, subjects the read second encrypted file key to the decryption algorithm D4 using the key information as a key to generate a file key, and outputs the file key to the decryption unit 305b via the switch unit 303b (step S244).

**[0122]** Following this, the decryption unit 305b reads a ciphertext included in the data part of the encrypted file 404b in the storage unit 400b, and subjects the read ciphertext to the decryption algorithm D3 using the file key as a key to generate a decrypted text (step S247), and writes a decrypted text file 402b including the generated decrypted text, to the storage unit 400b (step S248).

2.3 Conclusions

**[0123]** The file management apparatus 10b has the three functions : password registration; plaintext encryption; and ciphertext decryption.

**[0124]** For registering a password, the user loads the key storage medium storing key information beforehand, on the file management apparatus 10b, and inputs a password to be registered. The password registration unit 100b encrypts the input password using the key information, and stores the generated encrypted

password in the computer system. In the second embodiment, information to be encrypted and a key used in the encryption are reversed as compared with those in the first embodiment.

**[0125]** For encrypting a plaintext, the user first loads the key storage medium on the computer system in which the encrypted password is present, and designates a file to be encrypted. The file encryption unit 200b first decrypts the encrypted password using the key information so as to obtain a password. Following this, the file encryption unit 200b encrypts a generated file key using the password, to generate a first encrypted file key. Also, the file encryption unit 200b encrypts the file key using the key information to generate a second encrypted file key. The file encryption unit 200b then encrypts information stored in the file using the file key to generate a ciphertext, and writes an encrypted file including the first encrypted file key and the second encrypted file key in the header part thereof, and the ciphertext in the data part thereof, to the storage unit 400b.

**[0126]** For decrypting a ciphertext, there are two methods, one using key information and the other using a password. When using key information, the file decryption unit 300b decrypts the second encrypted file key acquired from the header part of the encrypted file 404b using the key information, to obtain a file key. The file decryption unit 300b then decrypts the ciphertext using the obtained file key as a key. When using a password, the file decryption unit 300b receives an input of the password from the user, decrypts the first encrypted file key using the received password to obtain a file key, and decrypts the ciphertext using the file key as a key to obtain the original plaintext.

2.4 Modification

**[0127]** Although the present invention has been described based on the second embodiment, the invention should not be limited to such. For instance, the file management apparatus 10b may be constructed according to the following modifications.

**[0128]** (1) The password registration unit 100b may further receive an input of a user ID that identifies the user, and store the encrypted password in association with the user ID, in a specific computer system. In this case, the file decryption unit 200b receives an input of a user ID, and then decrypts an encrypted password that is associated with the input user ID.

**[0129]** (2) The following is an explanation of the operation of the file management apparatus 10b when a password is changed, with reference to a flowchart shown in Fig. 14.

**[0130]** The password registration unit 100b reads key information from the key storage medium 20, reads a second encrypted file key from the encrypted file 404b, and subjects the second encrypted file key to the decryption algorithm D4 using the key information as a key to generate a file key (step S261). Following this, the password registration unit 100b receives an input of a new password from the user (step S262), subjects the generated file key to the encryption algorithm E2 using the new password as a key to generate a new first encrypted file key (step S263), and updates the first encrypted file key in the encrypted file 404b to the new first encrypted file key (step S264).

**[0131]** (3) For preventing encrypted information from being decrypted using a password, the file management apparatus 10b deletes the first encrypted file key in the encrypted file 404b. In this case, decryption using key information is available.

**[0132]** (4) The following is an explanation of the operation of the file management apparatus 10b when key information is updated, with reference to a flowchart shown in Fig. 15.

**[0133]** The key storage medium stores new key information beforehand, instead of the key information employed previously (referred to as old key information).

**[0134]** The file encryption unit 200b receives an input of a password that is the same as the password received previously (step S281), reads a first encrypted file key from the encrypted file 404b (step S282), and subjects the first encrypted file key to the decryption algorithm D2 using the received password as a key to generate a file key (step S283). Following this, the file encryption unit 200b reads the new key information from the key storage medium, subjects the file key to the encryption algorithm E4 using the new key information as a key to generate a new second encrypted file key (step S284), and updates the second encrypted file key in the encrypted file 404b to the new second encrypted file key (step S285).

**[0135]** (5) In the above embodiment, the encrypted password is stored in a computer system in which a plaintext has been encrypted to generate a ciphertext, and so decryption of the ciphertext using a password is made only possible within the computer system. To enable the decryption of the ciphertext using the password in another computer system, the encrypted key may be stored in a portable storage medium, and inputted into the other computer system.

**[0136]** Here, the password registration unit 100b in the computer system writes the encrypted password to a portable storage medium such as a SD memory card. Also, the user writes the encrypted file to another portable storage medium. The user then loads the portable storage medium to which the encrypted key has been written, and the portable storage medium to which the encrypted file has been written, on the other computer system, so that a file decryption unit in the other computer system reads the encrypted key from the portable storage medium, decrypts the read encrypted key, and also, reads the encrypted file from the portable storage medium, and decrypts the read encrypted file.

**[0137]** It should be noted here that the encrypted key and the encrypted file may be written to one portable

storage medium as separate files.

**[0138]** (6) When encrypting a plaintext to generate a ciphertext, the file encryption unit 200b may add various information to the header part of the encrypted file, the various information including encryption information indicating that the plaintext has been encrypted, and a user ID for the key information. In this case, when key information or a password is updated, the file encryption unit 200b may retrieve the encrypted file with reference to the additional information, such as encryption information indicating that the plaintext has been encrypted and a user ID for the key information, in procedures described in the items (2) or (4). Instead of writing such additional information to the header part of each encrypted file, the file encryption unit 200b may write such additional information for each encrypted file, to one unified file. In this case, the file encryption unit 200b retrieves each concerned encrypted file from the unified file in procedures described in the items (2) or (4).

**[0139]** (7) When encrypting a plaintext to generate a ciphertext, the file encryption unit 200b may further receive an input of a user indication, and determine whether to store a first encrypted file key into the header part of the encrypted file, according to the content of the user indication. When the first encrypted file key is determined to be stored, it is stored in the header part of the encrypted file as described above. When the first encrypted file key is determined not to be stored, neither generation nor storing of the first encrypted file key is performed. When the first encrypted file key is stored in the encrypted file, the ciphertext can be decrypted using a password. When the first encrypted file key is not stored in the encrypted file, the ciphertext is prohibited from being decrypted using a password.

**[0140]** (8) For prohibiting a ciphertext from being decrypted using key information in a case where the user loses the key information, the file management apparatus 10b deletes a second encrypted file key. This can prevent unauthorized users from decrypting encrypted information by acquiring the lost key information. In this way, the key information can be made temporarily invalid in the second embodiment, which is impossible in the first embodiment. In this case, decryption using a password is available.

**[0141]** Furthermore, according to the construction described in the item (4), the encrypted information can be decrypted using a password. Therefore, the user is allowed to have access to encrypted files without any inconvenience until new key information is issued. Also, when the new key information is issued, the only thing to do is to update the header part of each concerned encrypted file, so that decryption of each encrypted file using the new key information thereafter becomes possible.

**[0142]** The following is an explanation of operations when the user loses the key storage medium, with reference to flowcharts shown in Figs. 16 to 18.

**[0143]** As shown in these flowcharts, key information is made temporarily invalid when the user loses the key storage medium (step S301). When the user intends to decrypt a ciphertext while the key information is being invalid, a decryption process using a password is performed (step S302).

**[0144]** Next, new key information is issued. When the user is provided with a key storage medium storing the new key information, a new second encrypted file key is generated (step S303), and a normal decryption process is performed using the new key information (step S304).

**[0145]** The following explains detailed processes performed in steps S301 to S304.

**[0146]** In the process for making the key information temporarily invalid in Step S301, the file management apparatus 10b deletes the second encrypted file key (step S311).

**[0147]** In the decryption process using a password in step S302, the password input unit 301b receives an input of a password from the user (step S321), the decryption unit 302b reads the first encrypted file key from the storage unit 400b, subjects the read first encrypted file key to the decryption algorithm D2 using the password as a key to generate a file key, and outputs the generated file key to the decryption unit 305b via the switch unit 303b (step S322). Following this, the decryption unit 305b reads a ciphertext included in the data part of the encrypted file 404b in the storage unit 400b, and subjects the read ciphertext to the decryption algorithm D3 using the file key as a key to generate a decrypted text (step S323). The decryption unit 305b then writes the decrypted text file 402b including the generated decrypted text to the storage unit 400b (step S324).

**[0148]** In the new second encrypted file key generation process in step S303, the file encryption unit 200b receives an input of a password that is the same as the password received previously (step S331), reads the first encrypted file key from the encrypted file 404b (step S332), and subjects the first encrypted file key to the decryption algorithm D2 using the password as a key to generate a file key (step S333). Following this, the file encryption unit 200b reads new key information from the key storage medium, subjects the file key to the encryption algorithm E4 using the new key information as a key to generate a new second encrypted file key (step S334), and updates the second encrypted file key in the encrypted file 404b to the generated new second encrypted file key (step S335).

**[0149]** In the normal decryption process using the new key information in step S304, the decryption unit 304b reads the new key information from the key storage medium (step S341) and the new second encrypted file key from the storage unit 400b, subjects the read new second encrypted file key to the decryption algorithm D4 using the new key information as a key to generate a file key, and outputs the generated file key to the decryption unit 305b via the switch unit 303b (step S342). Following this, the decryption unit 305b reads a

ciphertext included in the data part of the encrypted file 404b in the storage unit 400b, subjects the read ciphertext to the decryption algorithm D3 using the file key as a key to generate a decrypted text (step S343), and writes the decrypted text file 402b including the generated decrypted text to the storage unit 400b (step S344).

**[0150]** (9) The file decryption unit 300b may require both key information and a password for decrypting a ciphertext.

**[0151]** Also, a first encrypted file key and a second encrypted file key each may be decrypted using both a password and key information, to generate two file keys, and an alteration in the header part of the encrypted file may be detected by judging whether the generated two file keys match or not.

**[0152]** (10) As in the first embodiment, authentication information may be added to an encrypted password, a first encrypted file key, a second encrypted file key, and a ciphertext, so that the authentication information can be utilized for detecting an alteration of each of the encrypted password, the first encrypted file key, the second encrypted file key, and the ciphertext.

3. Conclusions

**[0153]** According to the present invention as described above, encryption and decryption of a file using key information accompanying a computer becomes possible. In addition, decryption of the file only using a password that has been registered beforehand and stored securely in the computer is possible if indicated at the time of the encryption. The password does not need to be set each time a file is encrypted. Also, the present invention provides structures for making decryption using a password temporarily invalid, or easily changing the password, in case the user forgets the password. Also, the present invention further provides structures for making key information temporarily invalid in case the user loses the key information. When new key information is issued, a file that has encrypted with the lost key information can be decrypted using the new key information merely by updating the header part of the encrypted file. Also, by storing an ID for key information or for a password in a header part of each encrypted file or in a unified management file, each encrypted file that requires a change in accordance with updating key information or a password can be retrieved.

**[0154]** As described above, the present invention provides a file encryption/decryption system that satisfies the following conditions.

(1) Encryption of a file is performed using key information stored in a storage medium such as an IC card. Once a password is registered beforehand, it is not necessary to input a password every time encryption is performed.

(2) Decryption of a file is normally performed using the key information. Also, the decryption of the file using the password registered beforehand is made possible by a user indication at the time when the file is encrypted.

(3) The system comprises a structure allowing a password to be changed easily.

(4) The system comprises a structure that makes key information temporarily invalid when the key information is lost, a structure allowing, when new key information is issued, an encrypted file that has been encrypted using the key information, to be handled with the new key information, and a structure that easily retrieves an encrypted file to be changed due to the change of the key information.

4. Other Modifications

**[0155]** Although the present invention has been described based on the above embodiments, the invention should not be limited to such. For example, the following modifications are possible.

**[0156]** (1) In the above embodiments, DES is employed as the decryption/encryption algorithm. However, other decryption/encryption algorithms may instead be employed.

**[0157]** (2) The present invention also applies to the method used by the apparatuses described above. This method may be realized by computer programs that are executed by computers. Such computer programs may be distributed as digital signals.

**[0158]** Also, the present invention may be realized by a computer-readable storage medium, such as a floppy disk, a hard disk, a CD-ROM (Compact Disc-Read Only Memory), an MO (Magneto-Optical) disc, a DVD (Digital Versatile Disc), a DVD-ROM, a DVD-RAM, or a semiconductor memory, on which computer programs and/or digital signals mentioned above are recorded. Conversely, the present invention may also be realized by a computer program and/or digital signal that is recorded on a storage medium.

**[0159]** Computer program or digital signals that achieve the present invention may also be transmitted via a network, such as an electric communication network, a wired or wireless communication network, or the Internet.

**[0160]** Also, the above embodiments of the present invention can be realized by a computer system that includes a microprocessor and a memory. In this case, a computer program can be stored in the memory, with the microprocessor operating in accordance with the computer program.

**[0161]** The computer programs and/or digital signals may be provided on an independent computer system by distributing a storage medium on which the computer programs and/or digital signals are recorded, or by transmitting the computer programs and/or digital signals via a network. The independent computer may then execute the computer programs and/or digital signals to

function as the present invention.

**[0162]** (3) The limitations described in the embodiment and the modifications may be freely combined.

**[0163]** Although the present invention has been fully described by way of examples with reference to the accompanying drawings, it is to be noted that various changes and modifications will be apparent to those skilled in the art. Therefore, unless such changes and modifications depart from the scope of the present invention, they should be construed as being included therein.

## Claims

1. A file management apparatus that encrypts a plaintext to generate a ciphertext, stores the ciphertext, and decrypts the ciphertext, the file management apparatus comprising:

   a key storage medium storing key information beforehand;
   registration means for encrypting the key information using a password to generate an encrypted key;
   encryption means for encrypting a plaintext based on the key information to generate a ciphertext;
   switch means for switching between (a) generating key information by decrypting the encrypted key using the password and (b) reading the key information from the key storage medium; and
   decryption means for decrypting the ciphertext based on one of the generated key information and the read key information.

2. The file management apparatus of Claim 1 further comprising a memory unit,

   wherein the registration means receives an input of the password, encrypts the key information using the received password to generate the encrypted key, and writes the generated encrypted key to the memory unit,

   the encryption means encrypts the plaintext using a file key to generate the ciphertext, encrypts the file key using the key information to generate an encrypted file key, and writes the ciphertext in association with the encrypted file key, to the memory unit,
   the switch means

   (a) includes first key obtaining means for receiving an input of the password and decrypting the encrypted key using the received password to generate the key information, and second key obtaining means for reading the key information from the key storage medium, and
   (b) obtains the key information by one of the first key obtaining means and the second key obtaining means, and

   the decryption means decrypts the encrypted file key using the obtained key information to generate a file key, and decrypts the ciphertext using the file key to generate a decrypted text.

3. The file management apparatus of Claim 2,

   wherein the registration means further receives an input of a user identifier that identifies a user, and writes the user identifier in association with the encrypted key, to the memory unit, and
   the first key obtaining means further receives an input of the user identifier and decrypts the encrypted key that is associated with the user identifier.

4. The file management apparatus of Claim 2,

   wherein the registration means further writes the key information and/or authentication information in association with the encrypted key, to the memory unit,

   the encryption means further writes the encrypted key, the key information, and/or authentication information in association with the ciphertext, to the memory unit,
   the first key obtaining means checks, using the authentication information, whether the encrypted key has been altered or not, when the encrypted key that is associated with the authentication information is decrypted, and
   the decryption means checks, using the authentication information, whether the ciphertext has been altered or not, when the ciphertext that is associated with the authentication information is decrypted.

5. The file management apparatus of Claim 2,

   wherein the registration means writes the encrypted key to the memory unit that is a portable storage medium, and
   the first key obtaining means decrypts the encrypted key that has been written to the memory unit that is the portable storage medium.

6. The file management apparatus of Claim 2, further comprising

   deletion means for deleting the encrypted key that has been written to the memory unit.

7. The file management apparatus of Claim 2, further comprising

   deletion means for deleting the encrypted key

that has been written to the memory unit,

wherein the registration means further receives an input of a new password, encrypts the key information using the new password to generate a new encrypted key, and writes the generated new encrypted key to the memory unit.

8. The file management apparatus of Claim 2,

wherein the key storage medium stores new key information beforehand, instead of the key information,

the registration means receives the input of the password and decrypts the encrypted key using the password to generate key information, the encryption means decrypts the encrypted file key using the key information to generate a file key, encrypts the file key using the new key information to generate a new encrypted file key, and writes the new encrypted file key over the encrypted file key in the memory unit, and the registration means encrypts the new key information using the password to generate a new encrypted key and writes the new encrypted key over the encrypted key in the memory unit.

9. The file management apparatus of Claim 8,

wherein the registration means further receives an input of a user identifier that identifies a user,

the encryption means further writes the user identifier in association with the ciphertext and the encrypted file key, to the memory unit, and the encryption means retrieves the encrypted file key that is associated with the user identifier in the memory unit and generates a file key from the retrieved encrypted file key.

10. The file management apparatus of Claim 8,

wherein the encryption means further writes encryption information in association with the ciphertext and the encrypted file key, to the memory unit, the encryption information indicating that the plaintext has been encrypted, and

the encryption means retrieves the encrypted file key that is associated with the encryption information in the memory unit, and generates a file key from the retrieved encrypted file key.

11. The file management apparatus of Claim 8,

wherein the registration means further receives an input of a user identifier that identifies a user,

the encryption means further writes the user identifier in association with a file identifier that identifies the ciphertext and the encrypted file key, as a unified file, to the memory unit, and the encryption means extracts the file identifier that is associated with the user identifier from the unified file, specifies the encrypted file key identified by the extracted file identifier, and generates a file key from the specified encrypted file key.

12. The file management apparatus of Claim 8,

wherein the encryption means further writes encryption information in association with a file identifier that identifies the ciphertext and the encrypted file key, as a unified file, to the memory unit, the encryption information indicating that the plaintext has been encrypted, and

the encryption means extracts the file identifier that is associated with the encryption information from the unified file, specifies the encrypted file key identified by the extracted file identifier, and generates a file key from the specified encrypted file key.

13. The file management apparatus of Claim 2,

wherein the encryption means further writes the encrypted key in association with the ciphertext and the encrypted file key, to the memory unit, and

the first key obtaining means decrypts the encrypted key that is associated with the ciphertext and the encrypted file key.

14. The file management apparatus of Claim 13,

wherein the encryption means further receives an input of an indication, the indication showing whether the encrypted key and the ciphertext are to be written in association with each other to the memory unit, and writes, when the indication shows that the encrypted key and the ciphertext are to be written in association with each other, the encrypted key in association with the ciphertext, to the memory unit.

15. The file management apparatus of Claim 13,

wherein the registration means writes the generated encrypted key to the key storage medium instead of to the memory unit.

16. A file encryption apparatus that encrypts a plaintext to generate a ciphertext and stores the ciphertext into a memory unit thereof, the file management apparatus comprising:

a key storage medium storing key information beforehand;

registration means for receiving an input of a password, encrypts the key information using the received password to generate an encrypted key, and writes the generated encrypted key

to the memory unit; and
encryption means for encrypting a plaintext using a file key to generate a ciphertext, encrypting the file key using the key information to generate an encrypted file key, and writing the ciphertext in association with the encrypted file key, to the memory unit.

17. A file decryption apparatus that stores the ciphertext and the encrypted file key generated by the file encryption apparatus of Claim 16, in association with each other, in a memory unit thereof, and decrypts the ciphertext, the file decryption apparatus comprising:

a key storage medium storing key information beforehand;
switch means

(a) including first key obtaining means for receiving an input of a password and decrypting the encrypted key using the received password to generate key information, and second key obtaining means for reading the key information from the key storage medium, and
(b) obtaining the key information by one of the first key obtaining means and the second key obtaining means; and

decryption means for decrypting the encrypted file key using the obtained key information to generate a file key, and decrypts the ciphertext using the file key to generate a decrypted text.

18. A file management apparatus that encrypts a plaintext to generate a ciphertext, stores the ciphertext, and decrypts the ciphertext, the file management apparatus comprising:

a key storage medium storing key information beforehand;
registration means for encrypting a password using the key information to generate an encrypted password;
encryption means for encrypting a plaintext using a file key to generate a ciphertext, encrypting the file key based on a password obtained by decrypting the encrypted password to generate a first encrypted file key, and encrypting the file key based on the key information to generate a second encrypted file key;
switch means for switching between (a) decrypting the first encrypted file key based on the password and (b) decrypting the second encrypted file key based on the key information, to generate a file key; and
decryption means for decrypting the ciphertext using the generated file key.

19. The file management apparatus of Claim 18 further comprising a memory unit,

wherein the registration means receives an input of the password, encrypts the received password using the key information to generate the encrypted password, and writes the generated encrypted password to the memory unit,

the encryption means decrypts the encrypted password using the key information to generate the password, encrypts the plaintext using the file key to generate the ciphertext, encrypts the file key using the password to generate the first encrypted file key, encrypts the file key using the key information to generate the second encrypted file key, and writes the ciphertext in association with the first encrypted file key and the second encrypted file key, to the memory unit, the switch means

(a) includes first key obtaining means for receiving an input of the password and decrypting the first encrypted file key using the received password, and second key obtaining means for decrypting the second encrypted file key using the key information, and
(b) obtains the file key by one of the first key obtaining means and the second key obtaining means, and

the decryption means decrypts the ciphertext using the obtained file key to generate a decrypted text.

20. The file management apparatus of Claim 19,

wherein the registration means further receives an input of a user identifier that identifies a user, and writes the encrypted password in association with the user identifier, to the memory unit, and

the encryption means further receives an input of the user identifier and decrypts the encrypted password that is associated with the user identifier.

21. The file management apparatus of Claim 19,

wherein the encryption means receives an input of an indication, the indication showing whether the first encrypted file key is to be generated or not, and

(a) generates, when the indication shows that the first encrypted file key is to be generated, the first encrypted file key, and
(b) suppresses, when the indication shows that the first encrypted file key is not to be generated, both generating and writing of the first en-

crypted file key.

22. The file management apparatus of Claim 19,
wherein the registration means further writes authentication information in association with the encrypted password, to the memory unit,

the encryption means further checks, using the authentication information, whether the encrypted key has been altered or not, when the encrypted key is decrypted, and
the encryption means further writes the authentication information in association with each of the first encrypted file key, the second encrypted file key, and the ciphertext, to the memory unit,
the first key obtaining means and the second key obtaining means each check, using the authentication information associated with the first encrypted file key and the second encrypted file key, whether the first encrypted file key and the second encrypted file key have been altered or not, when the first encrypted file key and the second encrypted file key are decrypted, and
the decryption means checks, using the authentication information that is associated with the ciphertext, whether the ciphertext has been altered or not, when the ciphertext is decrypted.

23. The file management apparatus of Claim 19,
wherein the registration means writes the encrypted password to the key storage medium, instead of to the memory unit, and
the encryption means decrypts the encrypted password that has been written to the key storage medium.

24. The file management apparatus of Claim 19,
wherein the registration means further receives an input of a new password, encrypts the new password using the key information to generate a new encrypted password, and writes the generated new encrypted password over the encrypted password in the memory unit, and
the encryption means decrypts the second encrypted file key using the key information to generate a file key, encrypts the file key using the new password to generate a new first encrypted file key, and writes the new first encrypted file key over the first encrypted file key in the memory unit.

25. The file management apparatus of Claim 24,
wherein the registration means further receives an input of a user identifier that identifies a user,

the encryption means further writes the user identifier in association with the ciphertext, the first encrypted file key, and the second encrypted file key, to the memory unit, and
the encryption means retrieves the second encrypted file key that is associated with the user identifier, and decrypts the retrieved second encrypted file key.

26. The file management apparatus of Claim 24,
wherein the encryption means further writes encryption information in association with the ciphertext, the first encrypted file key, and the second encrypted file key, to the memory unit, the encryption information indicating that the plaintext has been encrypted, and
the encryption means retrieves the second encrypted file key that is associated with the encryption information, and decrypts the retrieved second encrypted file key.

27. The file management apparatus of Claim 24,
wherein the registration means further receives an input of a user identifier that identifies a user,

the encryption means further writes the user identifier in association with a file identifier that identifies the ciphertext, the first encrypted file key, and the second encrypted file key, as a unified file, to the memory unit, and
the encryption means extracts the file identifier that is associated with the user identifier from the unified file, specifies the second encrypted file key identified by the extracted file identifier, and decrypts the specified second encrypted file key.

28. The file management apparatus of Claim 24,
wherein the encryption means further writes encryption information in association with a file identifier that identifies the ciphertext, the first encrypted file key, and the second encrypted file key, as a unified file, to the memory unit, the encryption information indicating that the plaintext has been encrypted, and
the encryption means extracts the file identifier that is associated with the encryption information from the unified file, specifies the second encrypted file key identified by the extracted file identifier, and generates a file key from the specified second encrypted file key.

29. The file management apparatus of Claim 19 further comprising
deleting means for deleting the second encrypted file key that has been written to the memory unit.

30. The file management apparatus of Claim 19,
wherein the key storage medium stores new key information beforehand, instead of the key information,

the registration means receives the input of the password and decrypts the received password using the new key information to generate a new encrypted password, and writes the generated new encrypted password over the encrypted password in the memory unit, and
the encryption means decrypts the first encrypted file key using the password to generate a file key, encrypts the file key using the new key information to generate a new second encrypted file key, and writes the new second encrypted file key over the second encrypted file key in the memory unit.

31. The file management apparatus of Claim 30,
wherein the registration means further receives an input of a user identifier that identifies a user,

the encryption means further writes the user identifier in association with the ciphertext, the first encrypted file key, and the second encrypted file key, to the memory unit,
the encryption means retrieves the first encrypted file key that is associated with the user identifier and decrypts the retrieved first encrypted file key.

32. The file management apparatus of Claim 30,
wherein the encryption means further writes encryption information in association with the ciphertext, the first encrypted file key, and the second encrypted file key, to the memory unit, the encryption information indicating that the plaintext has been encrypted, and
the encryption means retrieves the first encrypted file key that is associated with the encryption information and decrypts the retrieved first encrypted file key.

33. The file management apparatus of Claim 30,
wherein the registration means further receives an input of a user identifier that identifies a user,

the encryption means further writes the user identifier in association with a file identifier that identifies the ciphertext, the first encrypted file key, and the second encrypted file key, as a unified file, to the memory unit, and
the encryption means extracts the file identifier that is associated with the user identifier from the unified file, specifies the first encrypted file key identified by the extracted file identifier, and decrypts the specified first encrypted file key.

34. The file management apparatus of Claim 30,
wherein the encryption means further writes encryption information in association with a file identifier that identifies the ciphertext, the first encrypted file key, and the second encrypted file key, as a unified file, to the memory unit, the encryption information indicating that the plaintext has been encrypted, and
the encryption means extracts the file identifier that is associated with the encryption information from the unified file, specifies the first encrypted file key identified by the extracted file identifier, and generates a file key from the specified first encrypted file key.

35. The file management apparatus of Claim 19,
wherein the switch means further receives an input of the password, decrypts the first encrypted file key using the received password to generate a first file key, decrypts the second encrypted file key using the key information to generate a second file key, judges whether the first file key and the second file key match, and detects an error when the first file key and the second file key do not match.

36. A file encryption apparatus that encrypts a plaintext to generate a ciphertext and stores the ciphertext in a memory unit thereof, the file encryption apparatus comprising:

a key storage medium storing key information beforehand;
registration means for receiving an input of a password, encrypts the received password using the key information to generate an encrypted password, and writes the generated encrypted password to the memory unit; and
encryption means for decrypting the encrypted password using the key information to generate a password, encrypts a plaintext using a file key to generate a ciphertext, encrypts the file key using the password to generate a first encrypted file key, encrypts the file key using the key information to generate a second encrypted file key, and writes the ciphertext in association with the first encrypted file key and the second encrypted file key, to the memory unit.

37. A file decryption apparatus that stores the ciphertext, the first encrypted file key, and the second encrypted file key generated by the file encryption apparatus of Claim 35, in association with each other, in a memory unit thereof, and decrypts the ciphertext, the file decryption apparatus comprising:

a key storage medium storing key information beforehand;
switch means

(a) including first key obtaining means for receiving an input of a password and decrypting the first encrypted file key using the received password, and second key obtaining means for decrypting the second encrypted file key using the key information, and
(b) obtaining a file key by one of the first key obtaining means and the second key obtaining means, and

decryption means for decrypting the ciphertext using the obtained file key to generate a decrypted text.

FIG. 1

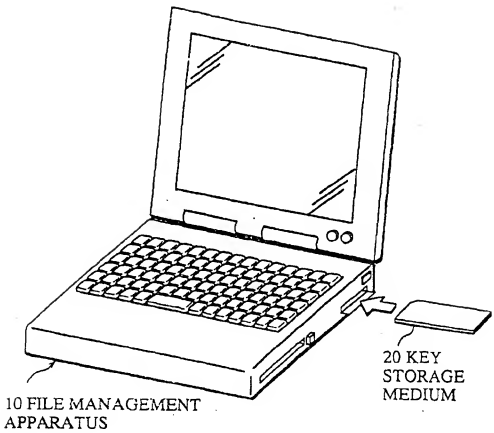


FIG. 2

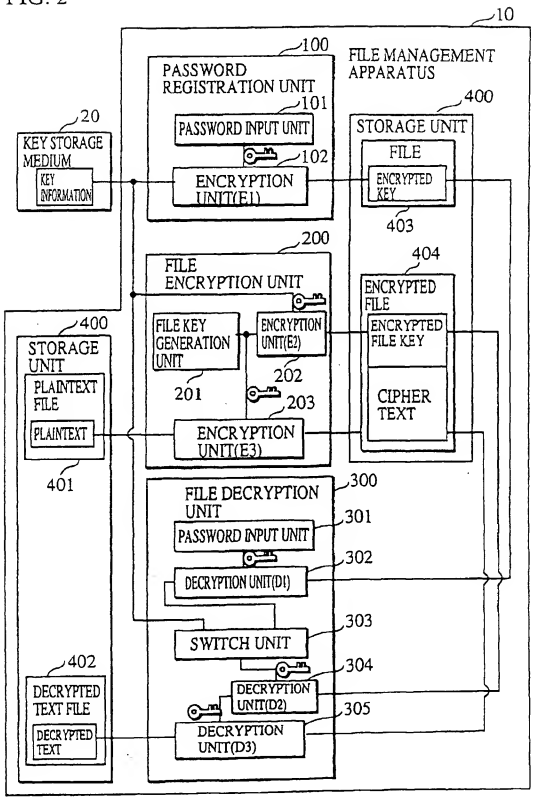

10
FILE MANAGEMENT APPARATUS
100
PASSWORD REGISTRATION UNIT
101
PASSWORD INPUT UNIT
102
ENCRYPTION UNIT(E1)
20
KEY STORAGE MEDIUM
KEY INFORMATION
400
STORAGE UNIT
FILE
ENCRYPTED KEY
403
404
ENCRYPTED FILE
ENCRYPTED FILE KEY
CIPHER TEXT
200
FILE ENCRYPTION UNIT
FILE KEY GENERATION UNIT
201
ENCRYPTION UNIT(E2)
202
203
ENCRYPTION UNIT(E3)
400
STORAGE UNIT
PLAINTEXT FILE
PLAINTEXT
401
300
FILE DECRYPTION UNIT
301
PASSWORD INPUT UNIT
302
DECRYPTION UNIT(D1)
303
SWITCH UNIT
304
DECRYPTION UNIT(D2)
305
DECRYPTION UNIT(D3)
402
DECRYPTED TEXT FILE
DECRYPTED TEXT

FIG. 3

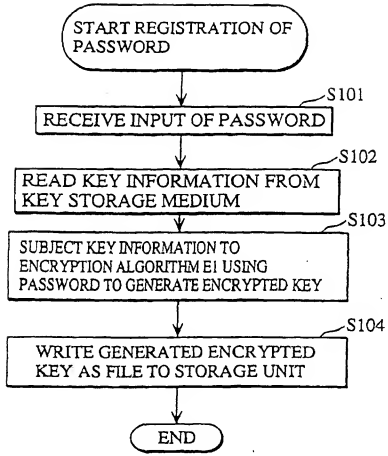

START REGISTRATION OF PASSWORD
S101
RECEIVE INPUT OF PASSWORD
S102
READ KEY INFORMATION FROM KEY STORAGE MEDIUM
S103
SUBJECT KEY INFORMATION TO ENCRYPTION ALGORITHM E1 USING PASSWORD TO GENERATE ENCRYPTED KEY
S104
WRITE GENERATED ENCRYPTED KEY AS FILE TO STORAGE UNIT
END

FIG. 4

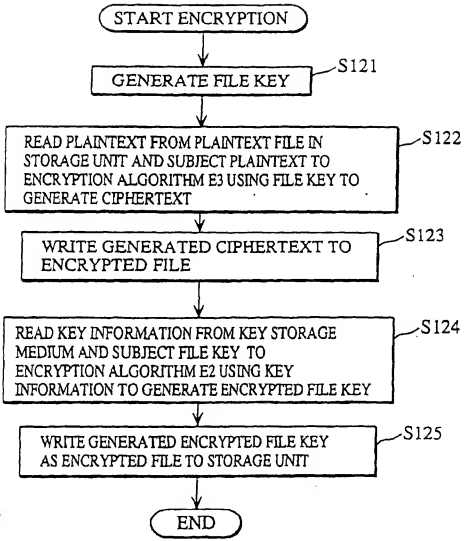

START ENCRYPTION
GENERATE FILE KEY
S121
READ PLAINTEXT FROM PLAINTEXT FILE IN STORAGE UNIT AND SUBJECT PLAINTEXT TO ENCRYPTION ALGORITHM E3 USING FILE KEY TO GENERATE CIPHERTEXT
S122
WRITE GENERATED CIPHERTEXT TO ENCRYPTED FILE
S123
READ KEY INFORMATION FROM KEY STORAGE MEDIUM AND SUBJECT FILE KEY TO ENCRYPTION ALGORITHM E2 USING KEY INFORMATION TO GENERATE ENCRYPTED FILE KEY
S124
WRITE GENERATED ENCRYPTED FILE KEY AS ENCRYPTED FILE TO STORAGE UNIT
S125
END

FIG. 5

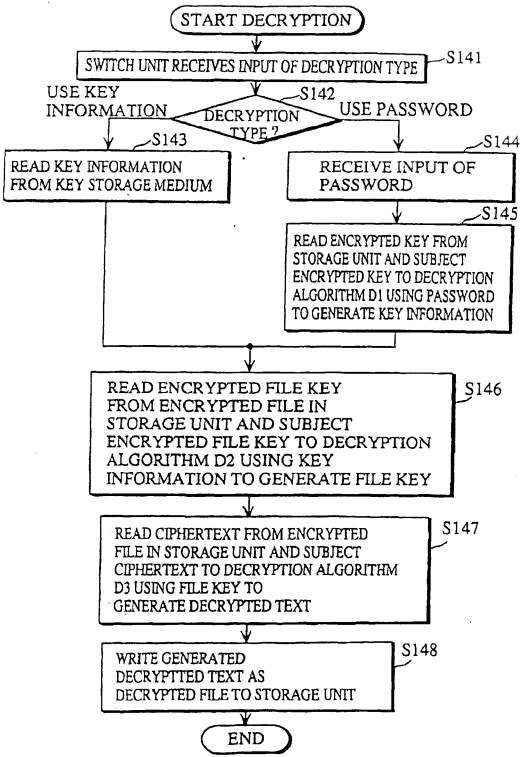

START DECRYPTION
SWITCH UNIT RECEIVES INPUT OF DECRYPTION TYPE
S141
USE KEY INFORMATION
DECRYPTION TYPE ?
S142
USE PASSWORD
S143
READ KEY INFORMATION FROM KEY STORAGE MEDIUM
S144
RECEIVE INPUT OF PASSWORD
S145
READ ENCRYPTED KEY FROM STORAGE UNIT AND SUBJECT ENCRYPTED KEY TO DECRYPTION ALGORITHM D1 USING PASSWORD TO GENERATE KEY INFORMATION
READ ENCRYPTED FILE KEY FROM ENCRYPTED FILE IN STORAGE UNIT AND SUBJECT ENCRYPTED FILE KEY TO DECRYPTION ALGORITHM D2 USING KEY INFORMATION TO GENERATE FILE KEY
S146
READ CIPHERTEXT FROM ENCRYPTED FILE IN STORAGE UNIT AND SUBJECT CIPHERTEXT TO DECRYPTION ALGORITHM D3 USING FILE KEY TO GENERATE DECRYPTED TEXT
S147
WRITE GENERATED DECRYPTTED TEXT AS DECRYPTED FILE TO STORAGE UNIT
S148
END

FIG. 6

USER ID TABLE

| USER ID | ENCRYPTED KEY |
|---|---|
| USER1 | ENCRYPTED KEY FOR USER 1 |
| USER2 | ENCRYPTED KEY FOR USER 2 |
| ⋮ | ⋮ |

FIG. 7

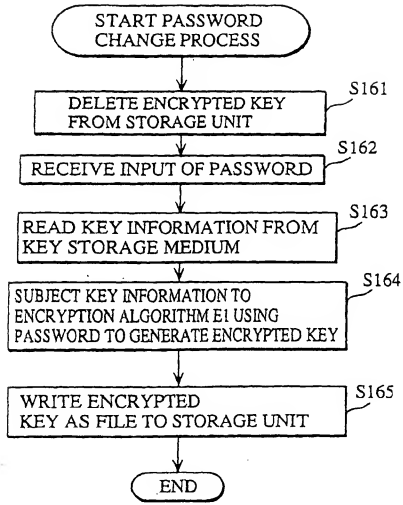

START PASSWORD CHANGE PROCESS
DELETE ENCRYPTED KEY FROM STORAGE UNIT
S161
RECEIVE INPUT OF PASSWORD
S162
READ KEY INFORMATION FROM KEY STORAGE MEDIUM
S163
SUBJECT KEY INFORMATION TO ENCRYPTION ALGORITHM E1 USING PASSWORD TO GENERATE ENCRYPTED KEY
S164
WRITE ENCRYPTED KEY AS FILE TO STORAGE UNIT
S165
END

FIG. 8

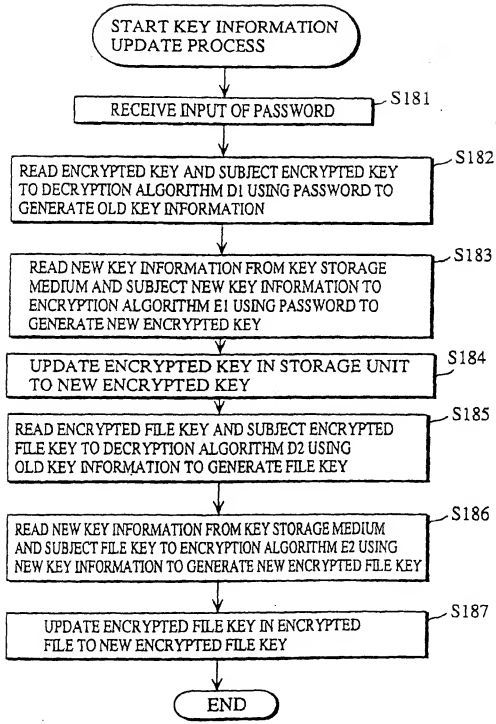

START KEY INFORMATION UPDATE PROCESS
RECEIVE INPUT OF PASSWORD
S181
READ ENCRYPTED KEY AND SUBJECT ENCRYPTED KEY TO DECRYPTION ALGORITHM D1 USING PASSWORD TO GENERATE OLD KEY INFORMATION
S182
READ NEW KEY INFORMATION FROM KEY STORAGE MEDIUM AND SUBJECT NEW KEY INFORMATION TO ENCRYPTION ALGORITHM E1 USING PASSWORD TO GENERATE NEW ENCRYPTED KEY
S183
UPDATE ENCRYPTED KEY IN STORAGE UNIT TO NEW ENCRYPTED KEY
S184
READ ENCRYPTED FILE KEY AND SUBJECT ENCRYPTED FILE KEY TO DECRYPTION ALGORITHM D2 USING OLD KEY INFORMATION TO GENERATE FILE KEY
S185
READ NEW KEY INFORMATION FROM KEY STORAGE MEDIUM AND SUBJECT FILE KEY TO ENCRYPTION ALGORITHM E2 USING NEW KEY INFORMATION TO GENERATE NEW ENCRYPTED FILE KEY
S186
UPDATE ENCRYPTED FILE KEY IN ENCRYPTED FILE TO NEW ENCRYPTED FILE KEY
S187
END

FIG. 9

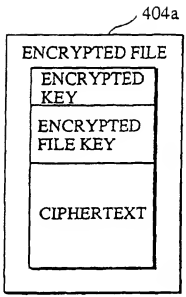

404a
ENCRYPTED FILE
ENCRYPTED KEY
ENCRYPTED FILE KEY
CIPHERTEXT

FIG. 10

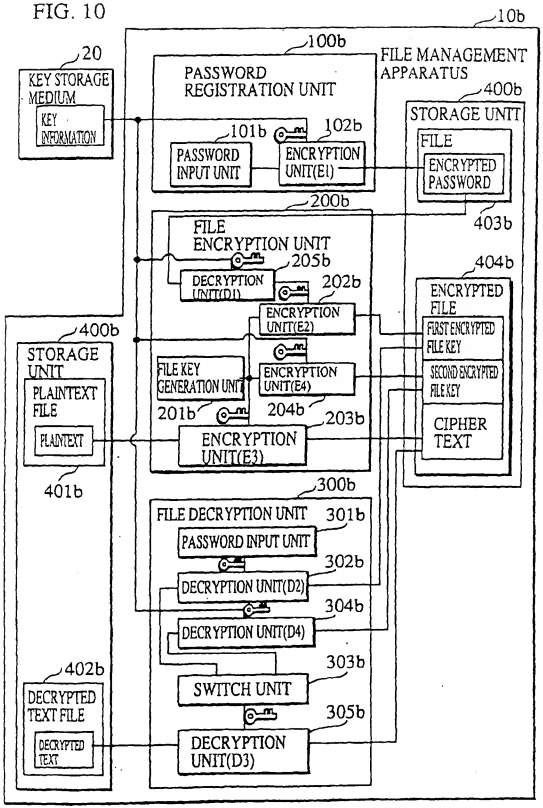

10b
FILE MANAGEMENT APPARATUS
20
KEY STORAGE MEDIUM
KEY INFORMATION
100b
PASSWORD REGISTRATION UNIT
101b
102b
PASSWORD INPUT UNIT
ENCRYPTION UNIT(E1)
400b
STORAGE UNIT
FILE
ENCRYPTED PASSWORD
403b
200b
FILE ENCRYPTION UNIT
205b
DECRYPTION UNIT(D1)
202b
ENCRYPTION UNIT(E2)
404b
ENCRYPTED FILE
FIRST ENCRYPTED FILE KEY
SECOND ENCRYPTED FILE KEY
CIPHER TEXT
400b
STORAGE UNIT
PLAINTEXT FILE
PLAINTEXT
401b
FILE KEY GENERATION UNIT
ENCRYPTION UNIT(E4)
201b
204b
203b
ENCRYPTION UNIT(E3)
300b
FILE DECRYPTION UNIT
301b
PASSWORD INPUT UNIT
302b
DECRYPTION UNIT(D2)
304b
DECRYPTION UNIT(D4)
303b
SWITCH UNIT
305b
402b
DECRYPTED TEXT FILE
DECRYPTED TEXT
DECRYPTION UNIT(D3)

FIG. 11

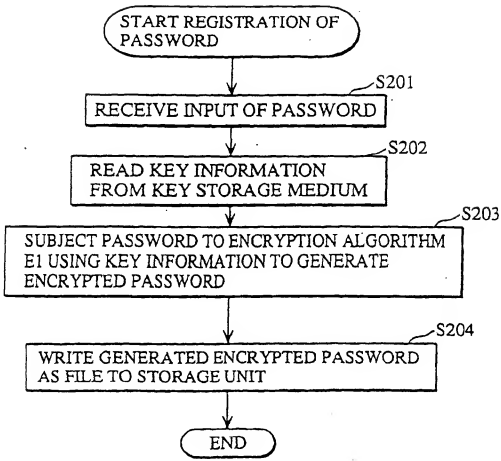

START REGISTRATION OF PASSWORD
S201
RECEIVE INPUT OF PASSWORD
S202
READ KEY INFORMATION FROM KEY STORAGE MEDIUM
S203
SUBJECT PASSWORD TO ENCRYPTION ALGORITHM E1 USING KEY INFORMATION TO GENERATE ENCRYPTED PASSWORD
S204
WRITE GENERATED ENCRYPTED PASSWORD AS FILE TO STORAGE UNIT
END

FIG. 12

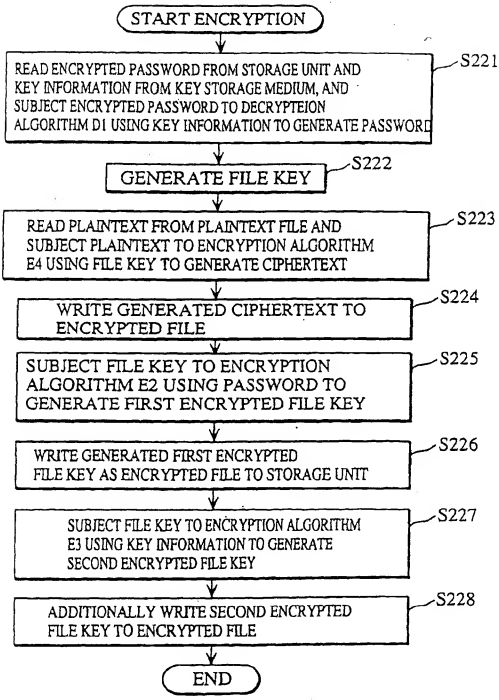

START ENCRYPTION
READ ENCRYPTED PASSWORD FROM STORAGE UNIT AND KEY INFORMATION FROM KEY STORAGE MEDIUM, AND SUBJECT ENCRYPTED PASSWORD TO DECRYPTEION ALGORITHM D1 USING KEY INFORMATION TO GENERATE PASSWORD
S221
GENERATE FILE KEY
S222
READ PLAINTEXT FROM PLAINTEXT FILE AND SUBJECT PLAINTEXT TO ENCRYPTION ALGORITHM E4 USING FILE KEY TO GENERATE CIPHERTEXT
S223
WRITE GENERATED CIPHERTEXT TO ENCRYPTED FILE
S224
SUBJECT FILE KEY TO ENCRYPTION ALGORITHM E2 USING PASSWORD TO GENERATE FIRST ENCRYPTED FILE KEY
S225
WRITE GENERATED FIRST ENCRYPTED FILE KEY AS ENCRYPTED FILE TO STORAGE UNIT
S226
SUBJECT FILE KEY TO ENCRYPTION ALGORITHM E3 USING KEY INFORMATION TO GENERATE SECOND ENCRYPTED FILE KEY
S227
ADDITIONALLY WRITE SECOND ENCRYPTED FILE KEY TO ENCRYPTED FILE
S228
END

FIG. 13

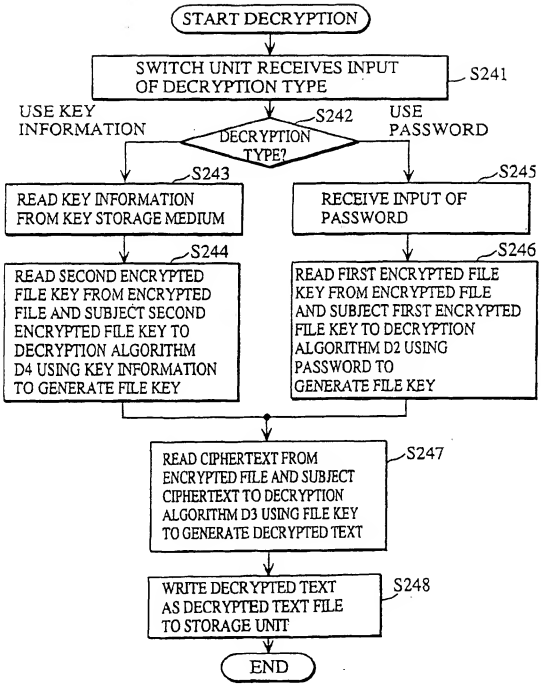

START DECRYPTION
SWITCH UNIT RECEIVES INPUT OF DECRYPTION TYPE
S241
USE KEY INFORMATION
DECRYPTION TYPE?
S242
USE PASSWORD
S243
READ KEY INFORMATION FROM KEY STORAGE MEDIUM
S245
RECEIVE INPUT OF PASSWORD
S244
READ SECOND ENCRYPTED FILE KEY FROM ENCRYPTED FILE AND SUBJECT SECOND ENCRYPTED FILE KEY TO DECRYPTION ALGORITHM D4 USING KEY INFORMATION TO GENERATE FILE KEY
S246
READ FIRST ENCRYPTED FILE KEY FROM ENCRYPTED FILE AND SUBJECT FIRST ENCRYPTED FILE KEY TO DECRYPTION ALGORITHM D2 USING PASSWORD TO GENERATE FILE KEY
READ CIPHERTEXT FROM ENCRYPTED FILE AND SUBJECT CIPHERTEXT TO DECRYPTION ALGORITHM D3 USING FILE KEY TO GENERATE DECRYPTED TEXT
S247
WRITE DECRYPTED TEXT AS DECRYPTED TEXT FILE TO STORAGE UNIT
S248
END

FIG. 14

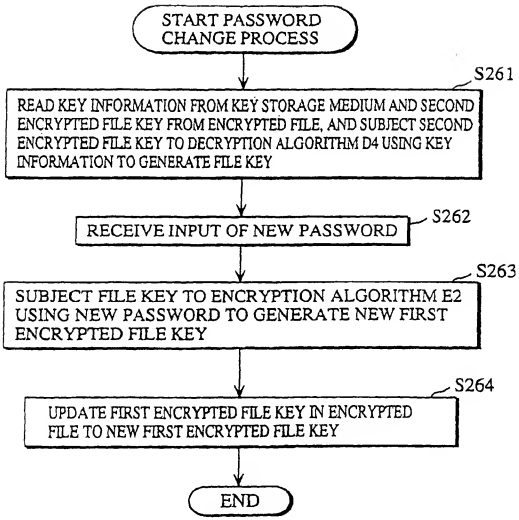

START PASSWORD CHANGE PROCESS
S261
READ KEY INFORMATION FROM KEY STORAGE MEDIUM AND SECOND ENCRYPTED FILE KEY FROM ENCRYPTED FILE, AND SUBJECT SECOND ENCRYPTED FILE KEY TO DECRYPTION ALGORITHM D4 USING KEY INFORMATION TO GENERATE FILE KEY
RECEIVE INPUT OF NEW PASSWORD
S262
S263
SUBJECT FILE KEY TO ENCRYPTION ALGORITHM E2 USING NEW PASSWORD TO GENERATE NEW FIRST ENCRYPTED FILE KEY
S264
UPDATE FIRST ENCRYPTED FILE KEY IN ENCRYPTED FILE TO NEW FIRST ENCRYPTED FILE KEY
END

FIG. 15

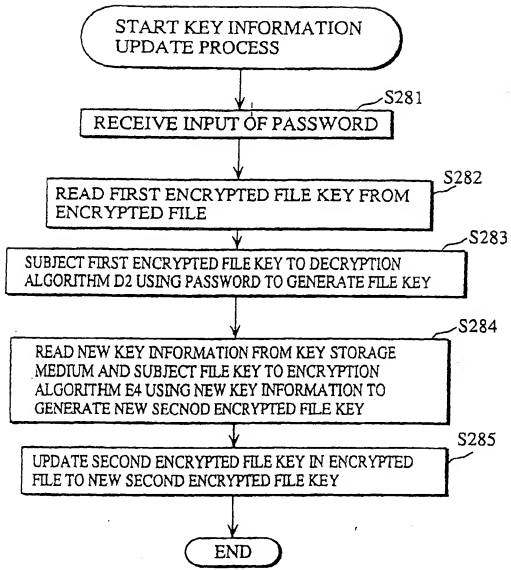

START KEY INFORMATION UPDATE PROCESS
S281
RECEIVE INPUT OF PASSWORD
S282
READ FIRST ENCRYPTED FILE KEY FROM ENCRYPTED FILE
S283
SUBJECT FIRST ENCRYPTED FILE KEY TO DECRYPTION ALGORITHM D2 USING PASSWORD TO GENERATE FILE KEY
S284
READ NEW KEY INFORMATION FROM KEY STORAGE MEDIUM AND SUBJECT FILE KEY TO ENCRYPTION ALGORITHM E4 USING NEW KEY INFORMATION TO GENERATE NEW SECNOD ENCRYPTED FILE KEY
S285
UPDATE SECOND ENCRYPTED FILE KEY IN ENCRYPTED FILE TO NEW SECOND ENCRYPTED FILE KEY
END

FIG. 16

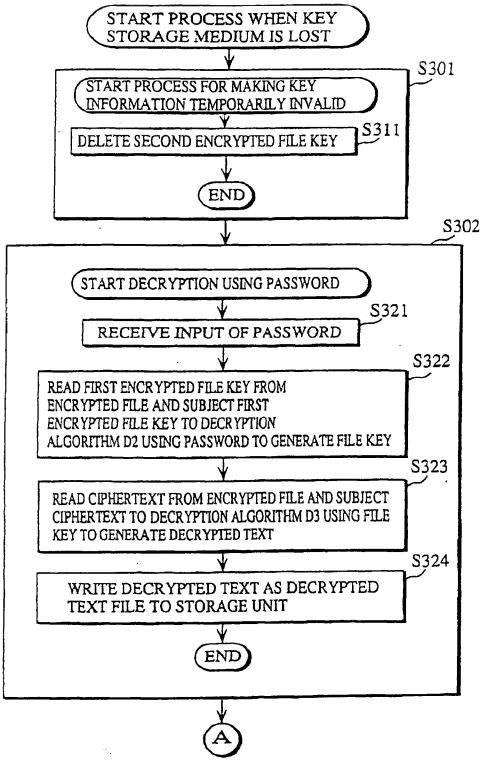

START PROCESS WHEN KEY STORAGE MEDIUM IS LOST
S301
START PROCESS FOR MAKING KEY INFORMATION TEMPORARILY INVALID
S311
DELETE SECOND ENCRYPTED FILE KEY
END
S302
START DECRYPTION USING PASSWORD
S321
RECEIVE INPUT OF PASSWORD
S322
READ FIRST ENCRYPTED FILE KEY FROM ENCRYPTED FILE AND SUBJECT FIRST ENCRYPTED FILE KEY TO DECRYPTION ALGORITHM D2 USING PASSWORD TO GENERATE FILE KEY
S323
READ CIPHERTEXT FROM ENCRYPTED FILE AND SUBJECT CIPHERTEXT TO DECRYPTION ALGORITHM D3 USING FILE KEY TO GENERATE DECRYPTED TEXT
S324
WRITE DECRYPTED TEXT AS DECRYPTED TEXT FILE TO STORAGE UNIT
END
A

FIG. 17

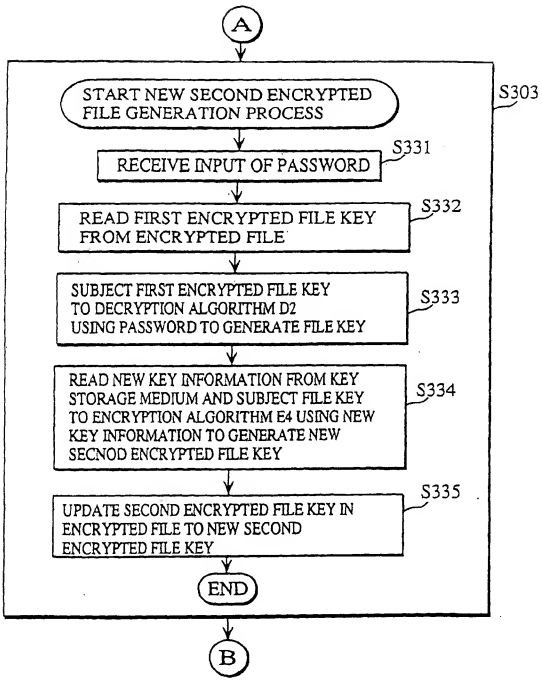
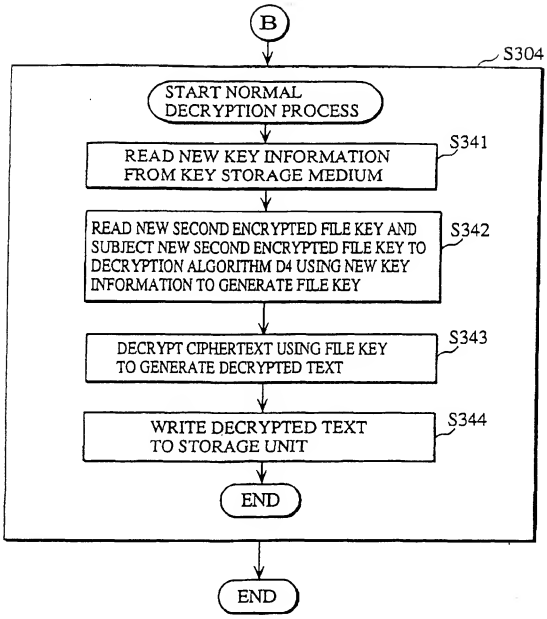

A
START NEW SECOND ENCRYPTED FILE GENERATION PROCESS
S303
RECEIVE INPUT OF PASSWORD
S331
READ FIRST ENCRYPTED FILE KEY FROM ENCRYPTED FILE
S332
SUBJECT FIRST ENCRYPTED FILE KEY TO DECRYPTION ALGORITHM D2 USING PASSWORD TO GENERATE FILE KEY
S333
READ NEW KEY INFORMATION FROM KEY STORAGE MEDIUM AND SUBJECT FILE KEY TO ENCRYPTION ALGORITHM E4 USING NEW KEY INFORMATION TO GENERATE NEW SECNOD ENCRYPTED FILE KEY
S334
UPDATE SECOND ENCRYPTED FILE KEY IN ENCRYPTED FILE TO NEW SECOND ENCRYPTED FILE KEY
S335
END
B

FIG. 18


B
S304
START NORMAL DECRYPTION PROCESS
READ NEW KEY INFORMATION FROM KEY STORAGE MEDIUM
S341
READ NEW SECOND ENCRYPTED FILE KEY AND SUBJECT NEW SECOND ENCRYPTED FILE KEY TO DECRYPTION ALGORITHM D4 USING NEW KEY INFORMATION TO GENERATE FILE KEY
S342
DECRYPT CIPHERTEXT USING FILE KEY TO GENERATE DECRYPTED TEXT
S343
WRITE DECRYPTED TEXT TO STORAGE UNIT
S344
END
END